



Clinical Reminders Index

PXRM*1.5*12

Technical Guide/Programmer's Manual

November 2004
Updated: December 2004

VISTA HSD&D
Department of Veterans Affairs

Revision History

Revision Date	Page or Chapter	Description
December 2004	Page 10	Removed Outpatient Pharmacy Clean-up Routine, per Pharmacy developers, who concluded that this routine could potentially cause more problems than it would fix.

Contents

Clinical Reminders Index Global 1

Introduction.....	1
Global Placement.....	1
Journaling.....	1
Estimating Global Size	2
Caché Sites.....	2
DSM Sites	3
Clinical Reminders Index Management.....	4
PXRM INDEX BUILD Option	4
Error Messages.....	5
Clean-up Recommendations	6
PXRM INDEX COUNT Option.....	8
Index Details	9
Inpatient Pharmacy	9
Outpatient Pharmacy.....	9
Order Entry	9
Lab	10
Mental Health.....	10
PCE	11
Problem List.....	12
Radiology	12
Registration	12
Vitals	12
Cross-References	14
LAB.....	16
Using FileMan to obtain detailed Cross-Reference descriptions.....	18

Clinical Reminders Index Global

Introduction

The Clinical Reminders index global has been designed to provide an index of clinical data, which supports rapid access to clinical data. It is used by Clinical Reminders v2.0, which evaluates reminders in 1/3 to 1/2 the time that v1.5 required. A large part of the speedup can be attributed to the index, because it provides such an efficient way to find patient data. The index is a resource that can be used by other packages or site-developed applications whenever they need to find patient data. Use of the index is supported by subscription to DBIA 4290. This document describes the structure of the index and how to use it, as well as how to populate it and manage it.

The basic structure of the index is:

```
^PXRMINDEX(FILE NUMBER,"IP",ITEM,DFN,DATE,DAS)
```

```
^PXRMINDEX(FILE NUMBER,"PI",DFN,ITEM,DATE,DAS)
```

where "IP" stands for item and patient and "PI" for patient and item.

The "IP" index lets you find all patients with a particular item, while the "PI" index lets you find all items for a patient. DAS stands for DA string, similar to FileMan's DA array. It is a semicolon-separated string that specifies the exact location in the global where the data is stored. This can be as simple as the IEN or can have a number of pieces (e.g., a lab test result has four pieces).

Indexes with items that are coded values, such as ICD 9 codes, vary from the basic structure, in order to support look-ups based on data such as primary diagnosis or principal procedure. Details are found in the sections describing each index.

Global Placement

This global serves as an index for the clinical data in a number of packages, so it is independent of a particular package. When new data is entered into the globals being indexed, the index will grow, so it needs to be placed where it has room to grow.

Journaling

Historical Seeding - Disable

Once the PXRMINDEX global is created, the historical seeding of the index can begin by using the options on the PXR INDEX MANAGEMENT menu. For this historical seeding, the Office of Information recommends that journaling of ^PXRMINDEX be **DISABLED**. During the initial creation and building of the index, journaling is not recommended because the index is not in use by the application and it can be rebuilt if necessary. In addition, there are concerns about journal files filling up with the large volume of SET commands that occur during the historical seeding of the PXRMINDEX global. Due to these factors, the Office of information recommends that journaling be disabled on the PXRMINDEX global until the historical index seeding is completed.

After Historical Seeding is completed - Enable

Once the options on PXR INDEX MANAGEMENT menu are completed, the Office of Information recommends that journaling of the PXRINDEX be **ENABLED**. The PXRINDEX index global will grow and will become utilized by Clinical Reminders Version 2.0 application when released. If the PXRINDEX global becomes corrupted, it can be rebuilt but the amount of time needed to rebuild the index will impact users of Clinical Reminders application. It is the recommendation of the Office of Information that if the index needs to be rebuilt, it is safer and more efficient to journal the index once the initial build and historical seeding are completed.

If you have any questions with respect to global placement or journaling, please contact the VSTS Team by either calling the VISTA Help Desk or by logging a Remedy ticket before proceeding with this installation.

Estimating Global Size

A utility for estimating the initial size of the index was distributed by patch PXR*1.5*20. To run this utility at the programmer prompt, type

D ESTTASK^PXRINDEX. This will start a TaskMan job that estimates the initial size of the index for each global as well as the total size. The information will be delivered in a MailMan message sent to members of the mail group defined in file #800. The estimated sizes will be given in blocks. These are 1K blocks for DSM sites and 2K blocks for Caché sites.

Caché/VMS Sites

The Office of Information recommends placing ^PXRINDEX in its own volume set. The initial size of the Caché dataset should be based on the estimated size, plus leaving room for significant growth.

Instructions for adding a new dataset can be obtained from the Avanti Library located at http://vaww.va.gov/custsvc/cssupp/axp/Cache_resource.asp.

Download "Adding a new dataset in Caché/VMS" (zip file)

Once the new dataset has been created and mounted, a new entry for the PXRINDEX global will need to be added to the Global Mapping section in the Caché Configuration Manager for each running Caché configuration and activated; e.g. Configuration Manager\Namespace\ VAH\Global Mapping.

NOTE: For Caché/VMS sites, you will need to update each running Caché configuration in the Cluster. It is IMPORTANT that each Caché configuration be updated; otherwise, unpredictable results can occur.

After the Global Mapping has been updated, create the PXRINDEX global:

- 1) Log into the VAH namespace
- 2) From the programmer mode prompt, issue the following command:

```
S ^PXRINDEX=""
```

It is highly recommended to perform the Caché steps without users on the system and PRIOR to the installation of the CLINICAL_REMINDERS_INDEX.KID build.

DSM Sites

It is recommended that ^PXRMINDEX be placed in its own volume set. The first volume, which is reserved for the DSM internal indexes, should be set at 2 GB instead of the usual 0.5 GB. The maximum supported size is 16 GB per volume set, so if the sizing tools estimate that the index global is going to be over 14 GB, then it should be split to begin with. You should also take into consideration the growth rate of the globals being indexed when deciding whether or not to split the global. You will need to leave enough room to accommodate the growth of the index global.

Clinical Reminders Index Management

When PXR*1.5*12 is installed, it creates a new option, PXR INDEX MANAGEMENT, which is a menu containing PXR INDEX BUILD and PXR INDEX COUNT.

The index building utility serves two purposes:

1. It initially populates the indexes by indexing the existing data. It works its way through the entire global, putting entries in the index for each piece of unique patient data it finds.
2. If the index ever gets corrupted or destroyed, the utility can be used to rebuild the index. Therefore, it isn't necessary to journal the index, since it can be recreated from the original data at any time. However, we do recommend journaling, as described in the Install Guide.

When the index utility finishes indexing a particular global, it sets the following three nodes:

```
^PXRMINDEX(FILE NUMBER,"GLOBAL NAME")=$$GET1^DID(FILE NUMBER,"","","GLOBAL NAME")
^PXRMINDEX(FILE NUMBER,"BUILT BY")=DUZ
^PXRMINDEX(FILE NUMBER,"DATE BUILT")=$$NOW^XLFTD
```

In addition to providing information about who built the index and when it got populated, these nodes can be used to determine when the index is complete and ready for use.

When data is added, edited, or deleted, the indexes are kept current using new-style Mumps cross-references. These cross-references call APIs that set and kill the index entries.

PXR INDEX BUILD Option

The index build utility can be accessed through the option PXR INDEX MANAGEMENT or directly via PXR INDEX BUILD.

```
Select OPTION NAME: PXR INDEX BUILD          PXR INDEX BUILD      run
routine
PXR INDEX BUILD
Which indexes do you want to (re)build?
  1 - LABORATORY TEST (CH, Anatomic Path, Micro)
  2 - MENTAL HEALTH
  3 - ORDER
  4 - PTF
  5 - PHARMACY PATIENT
  6 - PRESCRIPTION
  7 - PROBLEM LIST
  8 - RADIOLOGY
  9 - V CPT
 10 - V EXAM
 11 - V HEALTH FACTORS
 12 - V IMMUNIZATION
 13 - V PATIENT ED
 14 - V POV
 15 - V SKIN TEST
 16 - VITAL MEASUREMENT
Enter your list:  (1-16): 1-16
```

It can be used to build or rebuild the indexes for each of the globals, in any combination you want. You can run the build interactively or as a tasked job.

After selecting the globals to be indexed, you will be given the choice of submitting a tasked job or running it interactively. In either case, when the index build completes, members of the mail group defined in file #800 will be sent a MailMan message that looks something like:

```
Subj: Index for global AUPNVPED successfully built  [#12184]
10/03/02@10:25  5 lines
From: POSTMASTER (Sender: DOE,JOHN)  In 'IN' basket.    Page 1  *New*
-----
Build of Clinical Reminders index for global AUPNVPED completed.
Build finished at 10/03/2002@10:25:27
288 entries were created.
Elapsed time: 1 secs
0 errors were encountered.
```

NOTE: If the person who builds the indexes is not a member of the mail group, the messages will not be delivered to the members of the mail group, unless it is a public mail group.

For large globals, the build time can be many hours, so you may want to schedule these builds for overnight or weekends when the system is not under heavy use. The indexes for smaller globals, which require few block splits, can be built quickly (a few minutes), so you have a wider latitude in when to build those.

If you select a list of globals to be indexed, they will be indexed sequentially, and if it is tasked out, only one task will be created. If, on the other hand, you select a global and task it out and then go back and select another global and task it out, you could have two concurrent jobs both trying to write to PXRMINDEX at the same time. Some simple testing revealed that the total time for two concurrent jobs is longer than for two sequential jobs.

For example, we built the index for ORDERS and V CPT in a test account. Building them sequentially, ORDERS required 19:48 and V CPT required 19:19, for a total time of 1 15:07. When they were built concurrently, ORDERS required 1 15:58 and VCPT required 1 11:00, for a total time of 3 02:58, considerably longer than 1 15:07. Based on this data, the recommendation is to build the indexes sequentially, not concurrently.

Error Messages

If any entries couldn't be indexed, the completion message will look something like:

```
Subj: Index for global PS(55 successfully built [#12187] 10/03/02@10:30
6 lines
From: POSTMASTER (Sender: DOE,JOHN)  In 'IN' basket.    Page 1  *New*
-----
Build of Clinical Reminders index for global PS(55 completed.
Build finished at 10/03/2002@10:30:07
6416 entries were created.
Elapsed time: 13 secs
136 errors were encountered.
```

Another MailMan message contains the error information for up to the last N errors. The error information starts with the most recent entry in the global that has an error and progressively works back toward older entries. The number of errors displayed, N, is a site-configurable

parameter. The parameter is stored in the CLINICAL REMINDERS PARAMETERS file, #800, in the field MAXIMUM NUMBER OF INDEX ERRORS, field #15 of file #800, Clinical Reminder Parameters. The default value is 200. If you wish to change this, you can use the ENTER OR EDIT FILE ENTRIES FileMan option. If you find a substantial number of errors, it is likely there is a systematic problem, so after determining the cause and solution for the first few errors, you can probably apply the same corrective action to the bulk of them.

The message has the format: global, entry identification, and error message. The error message describes the problem – for example, missing or invalid data. Here is a sample of errors you might see for file #55:

```

Subj: CLINICAL REMINDER INDEX BUILD ERROR(S)  [#14895] 11/19/02@11:50 136 lines
From: POSTMASTER (Sender: DOE,JOHN) In 'IN' basket. Page 1
-----
GLOBAL: PS(55 ENTRY: DFN=1 D1=33 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=1 D1=34 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=1 D1=35 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=1 D1=36 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=2 D1=1 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=1719 Unit Dose missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=1 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=2 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=3 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=4 D1=2 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=4 D1=8 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=6 D1=5 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=6 D1=7 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=6 D1=590 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=8 D1=1 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=9 D1=2 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=9 D1=7 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=10 D1=31 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=10 D1=32 IV missing stop date

```

The entry information identifies the exact entry in the global that could not be indexed. If we examine the last line, it tells us that for ^PS(55,10,"IV",32,0), there is no stop date, so it can't be indexed. If you are not familiar with the structure of the global being indexed, it will be helpful to have a data dictionary listing on hand to help you interpret the entry identification information.

Entries that are not indexable will never be found by any application that uses the index to find data. Each site should make a decision concerning what they want to do about non-indexable entries. Some recommendations can be found below.

Clean-up Recommendations

Inpatient Pharmacy

Test sites reported that most of the errors are the same type: missing start date or missing patient.

Inpatient Pharmacy developers agree that there will be many of these. A former bug in the application caused those entries that have an E on the end. Sites can submit NOIS, if desired, and Inpatient RX developers will look at them.

```

GLOBAL: ^PS(55, ENTRY: DFN=994 D1=1545755 Unit Dose missing start date
Global ^PS(55,994,,1545755
      PS(55,994,,1545755

```

```

^PS(55,994,5,1545755,0) = ^^^^^^^E
^PS(55,994,5,1545755,9,0) = ^55.09D^1^1

GLOBAL: ^PS(55, ENTRY: DFN=388 D1=1804771 Unit Dose missing start date
Global ^PS(55,388,,1804771
      PS(55,388,,1804771
^PS(55,388,5,1804771,0) = ^^^^^^^E
^PS(55,388,5,1804771,9,0) = ^55.09D^1^1

GLOBAL: ^PS(55, ENTRY: DFN=1096 D1=1819001 Unit Dose missing start date
Global ^PS(55,1096,,1819001
      PS(55,1096,,1819001
^PS(55,1096,5,1819001,0) = ^^^^^^^E
^PS(55,1096,5,1819001,9,0) = ^55.09D^1^1
Global ^

GLOBAL: ^PS(55, ENTRY: DFN=1245 D1=16 IV missing start date
Global ^PS(55,1245,,16
      PS(55,1245,,16
^PS(55,1245,"IV",16,0) = 16^^^P^^^INFUSE VIA MINIPUMP^NOW^^1000^^^0^^S
X=PSSTA
TUS
^PS(55,1245,"IV",16,1) = ONE TIME
^PS(55,1245,"IV",16,2) = 2861123.1008^1^IBG
^PS(55,1245,"IV",16,3) = PROTECT FROM LIGHT/DO NOT REFRIGERATE
^PS(55,1245,"IV",16,6) = 67^20 MG
^PS(55,1245,"IV",16,"A",0) = ^55.04A^1^1
^PS(55,1245,"IV",16,"A",1,0) = 1^D^IBG^COMPUTER DOWN^2861123.1045
^PS(55,1245,"IV",16,"A",2,1,0) = ^55.15^1^1
^PS(55,1245,"IV",16,"A",2,1,1,0) = STATUS^DISCONTINUED^
^PS(55,1245,"IV",16,"AD",0) = ^55.02PA^1^1
^PS(55,1245,"IV",16,"AD",1,0) = 108^20 MG
^PS(55,1245,"IV",16,"SOL",0) = ^55.11IPA^1^1
^PS(55,1245,"IV",16,"SOL",1,0) = 1^10 ML

```

Outpatient Pharmacy

Pharmacy developers recommend that the entry should be skipped if certain data elements are missing that are required; e.g., dfn of patient, zero node, drug, etc. As noted above, entries that are non-indexable will automatically be skipped by applications using the index.

```

GLOBAL: ^PSRX( ENTRY: 5284873 missing days supply
Global ^PSRX(5284873
      PSRX(5284873
^PSRX(5284873,0) = 962688^24148^^^10069^^^^^^^^^^^^^1
^PSRX(5284873,2) = ^2981204^^^^^^^^^^
^PSRX(5284873,3) = 2981204
^PSRX(5284873,"OR1") = 464
^PSRX(5284873,"POE") = 1
^PSRX(5284873,"SIG") = ^0
^PSRX(5284873,"STA") = 11

GLOBAL: ^PSRX( ENTRY: 5287379 missing DFN
Global ^PSRX(5287379
      PSRX(5287379
^PSRX(5287379,0) = 5285540
^PSRX(5287379,2) = ^^^^^^^^^^^
^PSRX(5287379,3) =
^PSRX(5287379,"POE") = 1

GLOBAL: ^PSRX( ENTRY: 5288355 missing DFN

```

```
Global ^PSRX(5288355
      PSRX(5288355
^PSRX(5288355,0) = 5285807
^PSRX(5288355,2) = ^^^^^^^^^^^
^PSRX(5288355,3) =
^PSRX(5288355,"POE") = 1

GLOBAL: ^PSRX( ENTRY: 5285434 missing DFN
Global ^PSRX(5285434
      PSRX(5285434
^PSRX(5285434,0) = 963014
^PSRX(5285434,2) = ^^^^^^^^^^^
^PSRX(5285434,3) =
^PSRX(5285434,"POE") = 1
```

PXRM INDEX COUNT Option

The index count utility can be accessed through the option PXRM INDEX MANAGEMENT or directly via PXRM INDEX COUNT. This utility provides a count by year of the entries in the index. This will let sites see how their data is distributed on a yearly basis.

The selection for the count utility is identical to the build utility and the results are sent in a MailMan message just like the results of the build utility.

Index Details

Inpatient Pharmacy

The index is on file 55, Pharmacy Patient. The structure is:

`^PXRMINDEX(55,"IP",DRUG,DFN,START,STOP,DAS)`

`^PXRMINDEX(55,"PI",DFN,DRUG,START,STOP,DAS)`

where DRUG is a pointer to Drug file, START is the start date, and STOP is the stop date. The API to retrieve the associated data is `OEL^PSJPXRM1(DAS,.DATA)`.

It is documented in DBIA 3836. The API to retrieve the associated data is `OEL^PSJPXRM1`.

Non-VA Meds

The index is on the non-VA med multiple of file 55, Pharmacy Patient. The structure is:

`^PXRMINDEX("55NVA","IP",POI,DFN,START,STOP,DAS)`

`^PXRMINDEX("55NVA","PI",DFN,POI,START,STOP,DAS)`

where POI is a pointer to the Pharmacy Orderable Item file. START is the start date, if it exists; otherwise it is the documented date. STOP is the discontinued date if it exists; otherwise is "U"_D0. If STOP has the form "U"_D0 it should be interpreted to mean that no discontinued date exists so the patient is currently taking the non-VA med.

The API to retrieve the associated data is `NVA^PSOPXRM1(DAS,.DATA)`. It is documented in DBIA 3793.

Outpatient Pharmacy

The index is on file 52, Prescription file. The structure is:

`^PXRMINDEX(52,"IP",DRUG,DFN,START,STOP,DAS)`

`^PXRMINDEX(52,"PI",DFN,DRUG,START,STOP,DAS)`

where DRUG is a pointer to Drug file, START is the starting date (RELEASE DATE) and STOP is the stop date (RELEASE DATE + DAYS SUPPLY). The API to retrieve the associated data is `PSRX^PSOPXRM1(DAS,.DATA)`.

It is documented in DBIA 3793.

Order Entry

The index is on file 100, Order file. The structure is:

`^PXRMINDEX(100,"IP",OI,DFN,START,STOP,DAS)`

`^PXRMINDEX(100,"PI",DFN,OI,START,STOP,DAS)`

where OI is a pointer to the Orderable Items file, START is the START DATE, and STOP is the STOP DATE. The API to retrieve the associated data is `EN^ORX8(DA)`. Note that DA is the first piece of DAS and the data is returned in the array ORUPCHK.

The API is documented in DBIA 871.

Lab

The index is on file 63, Lab Data. The structure is:

Chemistry, Hematology, other Lab Tests

^PXRMINDEX(63,"IP",ITEM,DFN,DATE,DAS)

^PXRMINDEX(63,"PI",DFN,ITEM,DATE,DAS)

Microbiology and Anatomic Path data have an additional index

^PXRMINDEX(63,"PDI",DFN,DATE,ITEM,DAS)

where DATE is the Date/Time of collection. The structure of ITEM depends on the type of lab data.

For Chemistry, Hematology, and other lab tests, ITEM is numeric and a pointer to the Laboratory Test file.

For Microbiology, ITEM is of the format

"M:[S T O A M];#".

where the middle section can be one of:

S is specimen (# is a pointer to the Topography [SNOMED] file)

T is test (# is a pointer to the Laboratory Test file)

O is organism (# is a pointer to the Etiology Field [SNOMED] file)

A is antibiotic (# is a pointer to the Antimicrobial Susceptibility file)

M is a TB drug (# is the field number of the TB drug - ^DD(63.39,)).

For Anatomic Pathology, ITEM is of the format

"A:[S T O D M E F P I];#".

where the middle section can be one of:

S is specimen (# is 1.free text value)

T is test (# is a pointer to the Laboratory Test file)

O is organ/tissue (# is a pointer to the Topography [SNOMED] file)

D is disease (# is a pointer to Disease Field [SNOMED] file)

M is morphology (# is a pointer to the Morphology Field [SNOMED] file)

E is etiology (# is a pointer to the Etiology Field [SNOMED] file)

F is function (# is a pointer to the Function [SNOMED] file)

P is procedure (# is a pointer to the Procedure [SNOMED] file)

I is ICD9 (# is a pointer to the ICD DIAGNOSIS file)

Microbiology and Anatomic Pathology data are stored in a complex hierarchy. The ITEM is therefore a compound expression. This allows single elements of data to be easily found. The DAS also depends on the type of lab data. A chemistry test result has four semicolon pieces. Microbiology and Anatomic Pathology can be more complex and have a much more nested structure.

The API to retrieve the associated data is: LRPXRM^LRPXAPI(.DATA,DAS,ITEM,TYPE).

This information should be reviewed in the context of other data associated with the specimen.

The API is documented in DBIA 4245. The Lab package has other APIs that use these indexes.

Mental Health

The index is on file 601.2, Psych Instrument Patient. The structure is:

^PXRMINDEX(601.2,"IP",INS,DFN,DATE,DAS)

^PXRMINDEX(601.2,"PI",DFN,INS,DATE,DAS)

where INS is a pointer to the MH Instrument file. The API to retrieve the associated data is
ENDAS^YTAPI10(.YSDATA,DAS).

The API is documented in DBIA 4442.

PCE

There are indexes on all of the V files, with the exception of V Provider and V Treatment. There are two types of indexes for the V files – one for coded values, ICD 9 and CPT, and one for the non-coded values. The structure of the index for the non-coded values is:

^PXRMINDEX(file number,"IP",ITEM,DFN,DATE,DAS)

^PXRMINDEX(file number,"PI",DFN,ITEM,DATE,DAS)

where item is the .01 of the V file (for example, a pointer to the Education Topic file or Health Factor file), and DATE is the Visit Date.

The non-coded values are:

V FILE	FILE NUMBER
V EXAM	9000010.13
V HEALTH FACTORS	9000010.23
V IMMUNIZATION	9000010.11
V PATIENT ED	9000010.16
V SKIN TEST	9000010.12

The structure of the index for the coded values is

^PXRMINDEX(file number,"IPP",CODE,TYPE,DFN,DATE,DAS)

^PXRMINDEX(file number,"PPI",DFN,TYPE,CODE,DATE,DAS)

where CODE is a pointer to the coded value, TYPE is primary procedure for V CPT and primary/secondary for V POV. DATE is the Visit Date.

The coded values are:

V FILE	FILE NUMBER
V CPT	9000010.18
V POV	9000010.07

The APIs for retrieving the associated date are in routine PXPXRM. There is an entry point for each V file; these are listed in the following table. The APIs are documented in DBIA 4250.

V FILE	PXPXRM ENTRY POINT
V CPT	VCPT(DAS,.DATA)
V EXAM	VXAM(DAS,.DATA)
V HEALTH FACTORS	VHF(DAS,.DATA)
V IMMUNIZATION	VIMM(DAS,.DATA)
V PATIENT ED	VPEDU(DAS,.DATA)
V POV	VPOV(DAS,.DATA)
V SKIN TEST	VSKIN(DAS,.DATA)

Problem List

The index is on file 9000011, Problem File. The structure is:

^PXRMINDEX(9000011,"ISPP",ICD9,STATUS,PRIORITY,DFN,DLM,DAS)

^PXRMINDEX(9000011,"PSPI",DFN,STATUS,PRIORITY,ICD9,DLM,DAS)

where ICD9 is a pointer to the ICD Diagnosis file. STATUS is the status of the problem, either active ("A") or inactive ("I"). PRIORITY can be acute ("A"), chronic ("C", or null, in which case a "U" is stored. DLM is the Date Last Modified. This structure lets you quickly do things like find active problems whose status is acute. The API to retrieve the associated data is CALL2^GMPLUTL3.

The API is documented in DBIA 2644.

Radiology

The index is on file 70, Rad/Nuc Med Patient. The structure is:

^PXRMINDEX(70,"IP",PROC,DFN,DATE,DAS)

^PXRMINDEX(70,"PI",DFN,PROC,DATE,DAS)

where PROC is a pointer to the Rad/Nuc Med Procedures file. The API to retrieve the associated data is EN1^RAPXRM(DAS,.DATA).

The API is documented in DBIA 3731.

Registration

The index is on file 45, PTF. Because PTF stores both ICD 0 and ICD 9 codes, the index is structured to accommodate this.

^PXRMINDEX(45,"ICD0","INP",ICD0,DFN,DATE,NODE,DAS)

^PXRMINDEX(45,"ICD0","PNI",DFN,ICD0,DATE,NODE,DAS)

^PXRMINDEX(45,"ICD9","INP",ICD9,DFN,DATE,NODE,DAS)

^PXRMINDEX(45,"ICD9","PNI",DFN,ICD9,DATE,NODE,DAS)

where ICD0 is a pointer to the ICD Operation/Procedure file and ICD9 is a pointer to the ICD Diagnosis file. Because ICD 0 codes are stored at 10 different locations, five on the surgery node and five on the procedure node and ICD 9 codes are stored at 25 different locations, 10 on the discharge node, 5 on node 71, and 10 on the movement node, the storage node information is stored in the index. This allows quick retrieval from specific nodes. For example, one of the ICD 9 codes is stored as DXLS, so in this case the string "DXLS" is the node subscript. Some of the ICD 9 storage nodes have names like secondary diagnosis 1, secondary diagnosis 2 etc, the corresponding nodes would be D SD1, D SD1. Movement storage nodes have names like "M ICD2". The API to retrieve the associated data is PTF^DGPXRM(DAS,.DATA).

Note that for some of the cross-references in PTF, TYPE OF RECORD is included in the list of fields. It is used to exclude entries whose TYPE OF RECORD is CENSUS from the index.

The API is documented in DBIA #3817.

Vitals

The index is on file 120.5, GMV Vital Measurement. The structure is:

^PXRMINDEX(120.5,"IP",VITAL TYPE,DFN,DATE,DAS)

^PXRMINDEX(120.5,"PI",DFN,VITAL TYPE,DATE,DAS)

where VITAL TYPE is a pointer to the GMV Vital Type file. Entries that are marked as “entered-in-error” are not indexed. The API to retrieve the associated data is EN^GMVPXRM(.GMVDATA,DAS).

The API is documented in DBIA 3647.

Summary of the detailed index structure given above

Package	Structure	Pointer	API	DBIA
Inpatient Pharmacy	^PXRMINDEX(55 , "IP" , DRUG , DFN , START , STOP , DAS) ^PXRMINDEX(55 , "PI" , DFN , DRUG , START , STOP , DAS)	DRUG points to Drug file	OEL^PSJXPXRMI (DAS , .DATA)	3836
Lab	^PXRMINDEX(63 , "IP" , ITEM , DFN , DATE , DAS) ^PXRMINDEX(63 , "PI" , DFN , ITEM , DATE , DAS) ^PXRMINDEX(63 , "PDI" , DFN , DATE , ITEM , DAS)	ITEM is formatted depending on the type of data	LRPXRMLRPXAPI (.DATA , DAS , ITEM , TYPE)	4245
Mental Health	^PXRMINDEX(601.2 , "IP" , INS , DFN , DATE , DAS) ^PXRMINDEX(601.2 , "PI" , DFN , INS , DATE , DAS)	INS points to the MH Instrument file	ENDAS^YTAPI10 (.DATA , DAS)	4442
Non-VA Meds	^PXRMINDEX("55NVA" , "IP" , POI , DFN , START , STOP , DAS) ^PXRMINDEX("55NVA" , "PI" , DFN , POI , START , STOP , DAS)	POI points to the Pharmacy Orderable Item file	NVA^PSOPXRMI (DAS , .DATA)	3793
Order Entry	^PXRMINDEX(100 , "IP" , OI , DFN , START , STOP , DAS) ^PXRMINDEX(100 , "PI" , DFN , OI , START , STOP , DAS)	OI points to the Orderable Items file	EN^ORX8 (\$P (DAS , " ; "))	871
Outpatient Pharmacy	^PXRMINDEX(52 , "IP" , DRUG , DFN , START , STOP , DAS) ^PXRMINDEX(52 , "PI" , DFN , DRUG , START , STOP , DAS)	DRUG is a pointer to Drug file	PSRX^PSOPXRMI (DAS , .DATA)	3793
Problem List	^PXRMINDEX(9000011 , "ISPP" , ICD9 , STATUS , PRI , DFN , DLM , DAS) ^PXRMINDEX(9000011 , "PSPI" , DFN , STATUS , PRI , ICD9 , DLM , DAS)	ICD9 points to the ICD Diagnosis file	CALL2^GMPLUTL3 (DAS)	2644
Radiology	^PXRMINDEX(70 , "IP" , PROC , DFN , DATE , DAS) ^PXRMINDEX(70 , "PI" , DFN , PROC , DATE , DAS)	PROC points to the Rad/Nuc Med Procedures file	EN1^RAPXRMI (DAS , .DATA)	3731
Registration	^PXRMINDEX(45 , "ICD0" , "INP" , ICD0 , NUM , DFN , DATE , NODE , DAS) ^PXRMINDEX(45 , "ICD0" , "PNI" , DFN , NUM , IDC0 , DATE , NODE , DAS) ^PXRMINDEX(45 , "ICD9" , "INP" , ICD9 , DFN , NUM , DATE , NODE , DAS) ^PXRMINDEX(45 , "ICD9" , "PNI" , DFN , NUM , IDC9 , DATE , NODE , DAS)	ICD0 points to the ICD Operation/ Procedure file and ICD9 points to the ICD Diagnosis file.	PTF^DGPXRMI (DAS , .DATA)	3817
Vitals	^PXRMINDEX(120.5 , "IP" , VITAL TYPE , DFN , DATE , DAS) ^PXRMINDEX(120.5 , "PI" , DFN , VITAL TYPE , DATE , DAS)	VITAL TYPE points to the GMRV Vital Type file	EN^GMRVPXRMI (.DATA , DAS)	3647

Package	Structure	Pointer	API	DBIA
V Files (PCE)		ITEM is the .01 of the V file, for example a pointer to the Education Topic file or Health Factor file		4250
CPT	^PXRMINDEX(9000010.18,"IPP",CODE,TYPE,DFN,DATE,DAS) ^PXRMINDEX(9000010.18,"PPI",DFN,TYPE,CODE,DATE,DAS)		VCPT^PXPXRM(DAS,.DATA)	
Exam	^PXRMINDEX(9000010.13,"IP",ITEM,DFN,DATE,DAS) ^PXRMINDEX(9000010.13,"PI",DFN,ITEM,DATE,DAS)		VXAM^PXPXRM(DAS,.DATA)	
Health Factor	^PXRMINDEX(9000010.23,"IP",ITEM,DFN,DATE,DAS) ^PXRMINDEX(9000010.23,"PI",DFN,ITEM,DATE,DAS)	CODE is the coded value	VHF^PXPXRM(DAS,.DATA)	
Immunization	^PXRMINDEX(9000010.11,"IP",ITEM,DFN,DATE,DAS) ^PXRMINDEX(9000010.11,"PI",DFN,ITEM,DATE,DAS)	TYPE is the Principal Procedure	VIMM^PXPXRM(DAS,.DATA)	
Patient Ed	^PXRMINDEX(9000010.16,"IP",ITEM,DFN,DATE,DAS) ^PXRMINDEX(9000010.16,"PI",DFN,ITEM,DATE,DAS)		VPEDU^PXPXRM(DAS,.DATA)	
POV	^PXRMINDEX(9000010.07,"IPP",CODE,TYPE,DFN,DATE,DAS) ^PXRMINDEX(9000010.07,"PPI",DFN,TYPE,CODE,DATE,DAS)		VPOV^PXPXRM(DAS,.DATA)	
Skin Test	^PXRMINDEX(9000010.12,"IP",ITEM,DFN,DATE,DAS) ^PXRMINDEX(9000010.12,"PI",DFN,ITEM,DATE,DAS)		VSKIN^PXPXRM(DAS,.DATA)	

Cross-References

The index is kept current by using new-style FileMan cross-references that fire whenever data is added, edited, or deleted. The cross-references are added to the respective packages' data dictionaries when the multi-package build PXR*1.5*12 is installed. A list of the cross-references that are added follows.

NOTE: Some of the packages do direct sets of the data into their globals instead of using FileMan. In those cases, the routines where the data is set or killed have been modified to call the package API that does the set or kill of the index entry.

LAB

Lab results are stored in the Lab Data file #63. This is a very hierarchical file with a strong dependence on the data dictionary.

The Lab package makes programming calls to update the ^PXRMINDEX indexes. Chemistry-type data updates the indexes when results are verified. Anatomic Pathology and Microbiology update indexes when results are reported and/or released. Any changes to existing lab data update the indexes. All indexes are set using SLAB^LRPX and killed using KLAB^LRPX.

Routines

Chemistry-type data updates in a central routine.

LRVER3A Chemistry data are updated on verification and editing of verified data. All transactions go through LRVER3A, which stores the results and sets all the cross-references. This routine calls CHSET^LRPX.

LROC It is very rare but chemistry data may be purged during purging of old orders and accessions. This only happens on data that is corrupted and not reportable. This routine calls CHKILL^LRPX.

All Microbiology and Anatomic Pathology data are updated using the same routine, UPDATE^LRPXR. Adding, editing, or deleting data invokes this call. Results are compared before and after editing. Any change will update the indexes. This routine is called from several routines:

LRAPDA

LRAPDSR

LRMIEDZ

LRMIEDZ2

LRMISTF1

LRMIV

LRMIV1

LRMIV2

Anatomic Pathology uses these routines to change the indexes:

LRAPDA, LRAPDSR, LRAPM, LRAPMRL, LRAPRES

Microbiology uses these routines to change the indexes:

LRMIEDZ, LRMIEDZ2, LRMINEW1, LRMISTF1, LRMIV, LRMIV1, LRMIV2, LRMIVER1

Lab Indexes

^PXRMINDEX(63,"PI",DFN,ITEM,DATE,DAS)

This index is used for finding results of tests on a patient.

^PXRMINDEX(63,"IP", ITEM,DFN,DATE,DAS)

This index is used for finding patients that have specific lab results.

^PXRMINDX(63,"PDI",DFN,DATE,ITEM,DAS)

This index is only used for Microbiology and Anatomic Pathology and is used for finding results on a patient for a specific time period. Chemistry-type data does not require this because the data are already stored in a similar format in the Lab Data file. Micro and AP data use a compound structure for ITEM (the lab test or other coded result) and the "PDI" index provides a faster path to the results. Also, AP data is broken into four sections: Autopsy, Cytology, Electron Microscopy, and Surgical Pathology. This index collates results by collection date/time regardless of the section; again, making retrieval faster.

File #	File Name	Sub-file #	Sub-file Name	Cross-reference
45	PTF			ACR9DSD1 ACR9DSD10 ACR9DSD11 ACR9DSD12 ACR9DSD13 ACR9DSD14 ACR9DSD2 ACR9DSD3 ACR9DSD4 ACR9DSD5 ACR9DSD6 ACR9DSD7 ACR9DSD8 ACR9DSD9 ACR9DXLS ACR9PDX
		45.01	401	ACR0S1 ACR0S2 ACR0S3 ACR0S4 ACR0S5
		45.02	501	ACR9MICD1 ACR9MICD10 ACR9MICD2 ACR9MICD3 ACR9MICD4 ACR9MICD5 ACR9MICD6 ACR9MICD7 ACR9MICD8 ACR9MICD9
		45.05	601	ACR0P1 ACR0P2 ACR0P3 ACR0P4 ACR0P5
52	Prescription			ACRO
		52.1	Refill	ACRR

		52.2	Partial Date	ACRP
55	Pharmacy Patient	55.01	IV	ACRIV
		55.05	Non-VA meds	ACRNVA
		55.06	Unit Dose	ACRUD
63	Lab Data			None
70	Rad/Nuc Med Patient	70.03	Examinations	ACR
100	Order			None
120.5	GMV Vital Measurement			ACR
601.2	Psych Instrument Patient	601.22	Date	ACR
9000010.07	V POV			ACR
9000010.11	V IMMUNIZATION			ACR
9000010.12	V SKIN TEST			ACR
9000010.13	V EXAM			ACR
9000010.16	V PATIENT ED			ACR
9000010.18	V CPT			ACR
9000010.23	V HEALTH FACTORS			ACR
9000011	Problem			ACR

Using FileMan to obtain detailed Cross-Reference descriptions

If you wish a more detailed description of any of these cross-references, there are two different ways to get it – both use FileMan.

Method 1, Inquire Option

Use the Inquire option on the Index file.

VA FileMan 22.0

Select OPTION: **I** INQUIRE TO FILE ENTRIES

OUTPUT FROM WHAT FILE: INDEX//

Select INDEX: ACR

```

1  ACR    120.5  Clinical Reminders cross-reference.
2  ACR    70    Clinical Reminders index.
3  ACR    9000010.18  Clinical Reminders index.
4  ACR    9000010.23  Clinical Reminders index.
5  ACR    9000010.11  Clinical Reminders index.
```

Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-5:

```

6  ACR    9000010.16  Clinical Reminders index.
7  ACR    9000010.07  Clinical Reminders index.
8  ACR    9000010.12  Clinical Reminders index.
9  ACR    9000010.13  Clinical Reminders index.
10 ACR    601.2  Clinical Reminders cross-reference.
```

Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-10: 6 9000010.16 ACR Clinical Reminders index.

ANOTHER ONE:

STANDARD CAPTIONED OUTPUT? Yes// (Yes)

Include COMPUTED fields: (N/Y/R/B): NO// BOTH Computed Fields and Record
Number
(IEN)

NUMBER: 279
NAME: ACR

FILE: 9000010.16

```

SHORT DESCRIPTION: Clinical Reminders index.
TYPE: MUMPS                                EXECUTION: RECORD
ACTIVITY: IR                                ROOT TYPE: INDEX FILE
ROOT FILE: 9000010.16                       USE: ACTION
DESCRIPTION: This cross-reference builds two indexes. One for finding all
patients with a particular education topic and one for finding all the
education topics a patient has. The indexes are stored in the Clinical
Reminders index global as:
^PXRMINDX(9000010.16,"IP",EDUCATION TOPIC,DFN,VISIT DATE,DAS) and
^PXRMINDX(9000010.16,"PI",DFN,EDUCATION TOPIC,VISIT DATE,DAS) respectively.
SET LOGIC: D SVFILE^PXPXRM(9000010.16,.X,.DA)
KILL LOGIC: D KVFILE^PXPXRM(9000010.16,.X,.DA)
KILL ENTIRE INDEX CODE: K ^PXRMINDX(9000010.16)
ORDER NUMBER: 1                             TYPE OF VALUE: FIELD
FILE: 9000010.16                             FIELD: .01
SUBSCRIPT NUMBER: 1                         COLLATION: forwards
ORDER NUMBER: 2                             TYPE OF VALUE: FIELD
FILE: 9000010.16                             FIELD: .02
SUBSCRIPT NUMBER: 2                         COLLATION: forwards
ORDER NUMBER: 3                             TYPE OF VALUE: FIELD

```

Method 2, Data Dictionary Utility

The other way to obtain detailed cross-reference descriptions is to use the Data Dictionary Utility. You can look at an entire file or a sub-file. In the example below, we look at a sub-file.

VA FileMan 22.0

```

Select OPTION: DATA DICTIONARY UTILITIES
Select DATA DICTIONARY UTILITY OPTION: LIST FILE ATTRIBUTES
START WITH WHAT FILE: PTF//
GO TO WHAT FILE: PTF//
Select SUB-FILE: 601
Select LISTING FORMAT: STANDARD// INDEXES ONLY
What type of cross-reference (Traditional or New)? Both// NEW
Which field: ALL//
DEVICE: ANYWHERE Right Margin: 80//
NEW-STYLE INDEX LIST -- FILE #45                                08/18/04 PAGE 1
-----

```

Subfile #45.05

New-Style Indexes:

```

ACR0P1 (#450) RECORD MUMPS IR ACTION WHOLE FILE (#45)
Short Descr: Clinical Reminders index for ICD0 lookup.
Description: This cross-reference builds two indexes. One for finding
all patients with a particular ICD0 code and one for
finding all the ICD0 codes a patient has. The indexes are
stored in the Clinical Reminders index global as:
^PXRMINDX(45,"ICD0","INP",ICD0,NODE,DFN,DATE,DAS) and
^PXRMINDX(45,"ICD0","PNI",DFN,NODE,ICD0,DATE,DAS)
respectively. Date is the surgery/procedure date. NODE is
S followed by code number. For example, P1 means it was
found on the P node and it was operation code 1.
Set Logic: D SDGPT0^DGPTDDCR(.X,.DA,"P",1)
Kill Logic: D KDGP00^DGPTDDCR(.X,.DA,"P",1)
Whole Kill: K ^PXRMINDX(45,"ICD0")
X(1): PROCEDURE DATE (45.05,.01) (Subscr 1) (forwards)
X(2): PROCEDURE CODE 1 (45.05,4) (Subscr 2) (forwards)

```

The rest of the cross-references on this sub-file have been left out for brevity.

If you know the field number or field name of a field used in the cross-reference, you can select a single cross-reference for display.

VA FileMan 22.0

```
Select OPTION:      DATA DICTIONARY UTILITIES
Select DATA DICTIONARY UTILITY OPTION:      LIST FILE ATTRIBUTES
  START WITH WHAT FILE: PTF//
    GO TO WHAT FILE: PTF//
      Select SUB-FILE:
Select LISTING FORMAT: STANDARD// INDEXES ONLY
What type of cross-reference (Traditional or New)? Both// NEW
Which field: ALL//      SECONDARY DIAGNOSIS 1
DEVICE:      ANYWHERE      Right Margin: 80//
NEW-STYLE INDEX LIST -- FILE #45, FIELD #79.16                      08/18/04      PAGE 1
-----
```

```
ACR9DSD1 (#566)      RECORD      MUMPS      IR      ACTION
  Short Descr:      Clinical Reminders index for ICD9 lookup.
  Description:      This cross-reference builds two indexes. One for finding
                    all patients with a particular ICD9 code and one for
                    finding all the ICD9 codes a patient has. The indexes are
                    stored in the Clinical Reminders index global as:
                    ^PXRMINDEX(45,"ICD9","INP",ICD9,NAME,DFN,DATE,DAS) and
                    ^PXRMINDEX(45,"ICD9","PNI",DFN,NAME,ICD9,DATE,DAS)
                    respectively. Date is the discharge date, if it does not
                    exist then the admission date is used. NAME is the name of
                    the ICD9 code field. For example, D SD1, where D SD tells
                    us it is a discharge secondary diagnosis.
    Set Logic:      D SDGPT9D^DGPTDDCR(.X,.DA,"D SD1")
    Kill Logic:      D KDGPT9D^DGPTDDCR(.X,.DA,"D SD1")
    Whole Kill:      K ^PXRMINDEX(45)
                    X(1): PATIENT (45,.01) (Subscr 1) (forwards)
                    X(2): ADMISSION DATE (45,2) (Subscr 2) (forwards)
                    X(3): TYPE OF RECORD (45,11) (Subscr 3) (forwards)
                    X(4): SECONDARY DIAGNOSIS 1 (45,79.16) (Subscr 4) (forwards)
                    X(5): DISCHARGE DATE (45,70)
```

